CURSO PROFISSIONAL DE TÉCNICO DE SISTEMAS - INFORMÁTICA

Programação/UFCD 0784 **Correção – 1.ºL**

20/11/2024 O Professor Estagiário: João Brito

Introdução

O tema que estamos a trabalhar tem como objetivo identificar e criar funções na linguagem C. Pretende-se que os alunos compreendam a noção de subprogramas bem como que com a sua utilização é possível aplicar os princípios da programação estruturada e reescrever códigos de uma forma mais simples, ou seja através da divisão de problemas em componentes simples como meio de solução para resolver problemas mais complexos.

Objetivos Gerais

- Adquirir a noção de subprograma;
- Conhecer as regras de declaração de subprogramas;
- Conhecer as regras de execução de subprogramas;
- Utilizar corretamente parâmetros;
- Distinguir os diferentes tipos de subprogramas;
- Conhecer os mecanismos de utilização de bibliotecas de subprogramas.

Objetivos Específicos

- Conceber programas através da divisão de problemas
- Estimular o raciocínio lógico e o pensamento computacional
- Saber escolher e adequar as soluções tecnológicas aos problemas a resolver
- Estimular a reflexão, a observação e autonomia

Tarefas

- Ler, interpretar e responder a um conjunto de questões teóricas acerca de programação modular e funções em linguagem C.
- Leitura, interpretação e análise de um problema.
- Codificação da solução na linguagem de programação C, através da utilização do software de desenvolvimento Dev C++, devidamente indentada e comentada.

Enunciado da Tarefa

1. **Conceitos Básicos:**
   a) Diga o que entende por programação modular.
   b) Liste três vantagens de utilizar a programação modular.
2. **Funções:**
   a) Defina o conceito de função em linguagem C.
   b) Explique a diferença entre: função com retorno e função sem retorno.
   c) Escreva a estrutura básica de uma função em linguagem C.
   d) Diga o que entende por argumentos de função. Pode recorrer a um exemplo concreto durante a sua explicação.
   e) Indique qual a estrutura de um programa em linguagem C.
3. **Exercícios práticos recorrendo ao programa do Dev C++:**
   Crie um programa em linguagem C que permita ao utilizador optar por:
   - Função sem retorno que verifica se um determinado número à escolha do utilizador é par ou ímpar.
   - Função com retorno que calcule a idade media entre 2 alunos.

   Nota: O programa **termina** quando o utilizador digitar o número **3.**

   Sugestão: Pode criar um pequeno menu para o utilizador escolher a opção que pretende.

Entrega do trabalho realizado

O trabalho final (documento word com as respostas às questões teóricas, código em) deve ser compactado (utilizar Winzip ou Winrar), o nome do ficheiro deve indicar a data, os

nomes dos elementos do grupo, a respetiva turma e disciplina (exemplo: 24-11-2024 jose_teixeira_e_bruno_caetano, 1.ºL, Programacao.zip).

O trabalho deverá ser publicado na tarefa criada na disciplina de Programação, na plataforma *Microsoft Teams* por todos os elementos do grupo até ao final da aula.

Duração da Tarefa

Uma sessão de 50 minutos + os primeiros 20 minutos de uma segunda sessão. Os últimos 20 minutos da segunda sessão serão para a correção da ficha de trabalho.

A tarefa deverá ser resolvida em grupos de dois ou três alunos, de acordo com a orientação do professor.

Avaliação da Tarefa

Esta tarefa tem carácter formativo

Correção da Ficha

O que entende por programação modular?

A programação modular é uma técnica de desenvolvimento de software que divide um programa em pequenas partes ou módulos independentes e reutilizáveis, cada um responsável por uma funcionalidade específica.

Três vantagens de utilizar a programação modular:

- Facilita a manutenção do código.
- Promove a reutilização de código em diferentes partes do programa ou em projetos futuros.
- Simplifica o processo de depuração, uma vez que os módulos podem ser testados individualmente.

Defina o conceito de função em linguagem C:

Uma função em C é um bloco de código reutilizável que executa uma tarefa específica e pode ser chamada em qualquer parte do programa.

Diferença entre função com retorno e função sem retorno:

- Função **com retorno**: Retorna um valor para o programa que a chamou. Exemplo: int soma(int a, int b) { return a + b; }.
- Função **sem retorno**: Não retorna nenhum valor, apenas executa ações. Exemplo: void imprimirMensagem() { printf("Olá Mundo!"); }.

Estrutura básica de uma função em C:

c

Copiar código

```
tipo_de_retorno nome_da_funcao(tipo_de_parametro parametro) {
    // Corpo da função
}
```

O que entende por argumentos de função?

Os argumentos são valores passados para uma função quando ela é chamada, para que possam ser utilizados dentro dela.

Exemplo:

```
int soma(int a, int b) {
    return a + b;
}
```

Estrutura básica de um programa em C:

```
#include <stdio.h>

int main() {
    // Código principal
    return 0;
}
```

Resolução dos exercícios práticos:

Titulo1

```c
#include <stdio.h>

void verificarParImpar(int numero); // Função sem retorno
float calcularIdadeMedia(int idade1, int idade2); // Função com retorno

int main() {
    int opcao;

    do {
        // Menu de opções
        printf("\n=== Menu de Opções ===\n");
        printf("1. Verificar se um número é par ou ímpar (Função sem retorno)\n");
        printf("2. Calcular a idade média entre dois alunos (Função com retorno)\n");
        printf("3. Sair\n");
        printf("Escolha uma opção: ");
        scanf("%d", &opcao);
```

```c
        switch(opcao) {
            case 1: {
                int numero;
                printf("Digite um número: ");
                scanf("%d", &numero);
                verificarParImpar(numero); // Chama a função sem retorno
                break;
            }
            case 2: {
                int idade1, idade2;
                printf("Digite a idade do primeiro aluno: ");
                scanf("%d", &idade1);
                printf("Digite a idade do segundo aluno: ");
                scanf("%d", &idade2);
                float media = calcularIdadeMedia(idade1, idade2); // Chama a função com retorno
                printf("A idade média é: %.2f\n", media);
                break;
            }
            case 3:
                printf("Programa encerrado.\n");
                break;
            default:
                printf("Opção inválida! Tente novamente.\n");
        }
    } while(opcao != 3); // Sai do programa quando o utilizador digita 3

    return 0;
}

// Função sem retorno para verificar se um número é par ou ímpar
void verificarParImpar(int numero) {
    if (numero % 2 == 0) {
        printf("O número %d é par.\n", numero);
    } else {
        printf("O número %d é ímpar.\n", numero);
    }
}

// Função com retorno para calcular a idade média entre dois alunos
float calcularIdadeMedia(int idade1, int idade2) {
    return (idade1 + idade2) / 2.0;
}
```